Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti

Facultatea de Electronică, Telecomunicaţii şi

Tehnologia Informaţiei

FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	**Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti**
1.2 Facultatea	**Electronică, Telecomunicaţii şi Tehnologia Informaţiei**
1.3 Departamentul	**Electronică Aplicată şi Ingineria Informaţiei**
1.4 Domeniul de studii	Calculatoare şi Tehnologia Informaţiei
1.5 Ciclul de studii	Masterat
1.6 Specializarea	Sisteme Inteligente şi Vedere Artificială

2. Date despre disciplină

2.1 Denumirea disciplinei (ro) (en)				Arhitectura informaţiei			
2.2 Titularul activităţilor de curs				S.l./Lect. Dr. George Valentin STOICA			
2.3 Titularul activităţilor de seminar / laborator				S.l./Lect. Dr. George Valentin STOICA			
2.4 Anul de studiu	1	2.5 Semestrul	II	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Ob
2.8 Tipul disciplinei		F	2.9 Codul disciplinei	4		2.10 Tipul de notare	Nota

3. Timpul total estimat (ore pe semestru al activităţilor didactice)

3.1 Număr de ore pe săptămână	2.5	Din care: 3.2 curs	1.50	3.3 seminar/laborator	1
3.4 Total ore din planul de învăţământ	35.00	Din care: 3.5 curs	21	3.6 seminar/laborator	14
Distribuţia fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie şi notiţe Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate Pregătire seminarii/ laboratoare/proiecte, teme, referate, portofolii şi eseuri					34
Tutorat					0
Examinări					6
Alte activităţi (dacă există):					0

3.7 Total ore studiu individual	40.00
3.8 Total ore pe semestru	75
3.9 Numărul de credite	3

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	Nu este cazul
4.2 de rezultate ale învăţării	Acumulare cunoştinţe de Structuri de date si algoritmi Programare obiect-orientata Sisteme de operare Calcul paralel

5. Condiţii necesare pentru desfăşurarea optimă a activităţilor didactice (acolo unde este cazul)

5.1 Curs	Cursul se va desfăşura într-o sală dotată cu videoproiector.
5.2 Seminar/ Laborator/Proiect	Laboratorul se va desfăşura într-o sală cu dotare specifică, ce trebuie să includă: pc-uri, mediu de dezvoltare software precum MS Visual Studio, GPU precum nVidia GTX

6. Obiectiv general *(Se referă la intenţiile profesorilor pentru studenţi, la ceea ce studenţii vor fi învăţaţi în timpul cursului. Oferă o orientare cu privire la locul cursului în cadrul domeniului ştiinţific abordat, precum şi la rolul pe care acesta îl are în cadrul specializării studiate. Vor fi descrise de o manieră generală tematicile abordate, justificarea includerii cursului în planul de învăţământ al specializării studiate etc.)*

Cursul are ca obiectiv aprofundarea de catre studenti a conceptelor cu privire la limbajele, structurile si arhitecturile de calcul de performanta utilizate in domeniul tehnologiei informatiei. Cursul prezinta sintetic limbajele utilizate in platformele de prelucrare a informatiei precum C/C++/Java/Python/.Net si continua cu arhitecturile de calcul paralele si distribuite cloud si GPU. Partea a doua a cursului aprofundeaza arhitectura GPU si mai concret arhitectura CUDA ca si platforma de calcul de inalta performanta.
Aplicatiile de laborator au ca obiectiv evidentierea practica a conceptelor si notiunilor specifice calculului si procesarii informatiei. Folosind drept modalitati de experimentare dezvoltarea de aplicatii CUDA, fie folosind tehnologii precum OpenCL, studentii isi pot insusi si pot experimenta cunostintele dobandite in timpul cursului. Intreg setul de notiuni dobandite in timpul cursului pot fi experimentate in timpul aplicatiilor de laborator

7. Competenţe *(Capacitatea dovedită de a utiliza cunoştinţe, aptitudini şi abilităţi personale, sociale şi/sau metodologice în situaţii de muncă sau de studiu şi pentru dezvoltarea profesională şi personală. Reflectă cerinţele angajatorilor.)*

Specifice	Dupa absolvirea acestui curs studentii isi insusesc notiunile de baza si avansate din domeniul programarii utilizand arhitecturi GPU/CUDA. Aceste cunostinte combinate cu parcurgerea aplicatiilor de laborator confera posibilitatea de a dezvolta aplicatii de prelucrare a informatiei. De asemenea, studentii vor fi capabili sa dezvolte tehnici si metode de programare, optimizare si eficientizare a algoritmilor de procesare a informatiei.
Transversale (generale)	Îndeplinirea sarcinilor profesionale cu identificarea exactă a obiectivelor de realizat, a unor factori potenţiali de risc, a resurselor disponibile, a aspectelor economico-financiare, condiţiilor de finalizare a acestora, etapelor de lucru, timpului de lucru şi termenelor de realizare aferente;

8. Rezultatele învăţării *(Sunt enunţuri sintetice referitoare la ceea ce un student va fi capabil să facă sau să demonstreze la finalizarea unui curs. Rezultatele învăţării reflectă realizările studentului şi mai puţin intenţiile profesorului. Rezultatele învăţării informează studenţii despre ceea ce se aşteaptă de la ei din punct de vedere al performanţei, pentru a obţine notele şi creditele dorite. Sunt definite în termeni concreţi, folosind verbe similare exemplelor de mai jos şi indică ceea ce se va urmări prin evaluare. Rezultatele învăţării vor fi astfel redactate încât să fie evidenţiată clar relaţia faţă de competenţele definite la punctul 7.)*

Cunoştinţe	*Rezultatul asimilării de informaţii prin învăţare. Cunoştinţele reprezintă ansamblul de fapte, principii, teorii şi practici legate de un anumit domeniu de muncă sau de studiu. Pot fi teoretice şi/sau faptice.* Dupa absolvirea acestui curs studentii isi insusesc notiunile de baza si avansate din domeniul programarii utilizand arhitecturi GPU/CUDA. Aceste cunostinte combinate cu parcurgerea aplicatiilor de laborator confera posibilitatea de a dezvolta aplicatii de prelucrare a informatiei. De asemenea, studentii vor fi capabili sa dezvolte tehnici si metode de programare, optimizare si eficientizare a algoritmilor de procesare a informatiei.

Aptitudini	*Capacitatea de a aplica cunoștințe și de a utiliza know-how pentru a duce la îndeplinire sarcini și a rezolva probleme. Aptitudinile sunt descrise ca fiind cognitive (implicând utilizarea gândirii logice, intuitive și creative) sau practice (implicând dexteritate manuală și utilizarea de metode, materiale, unelte și instrumente).* Selectează și grupează informații relevante într-un context dat.
Responsabilitate și autonomie	*Capacitatea cursantului de a aplica în mod autonom și responsabil cunoștințele și aptitudinile sale.* Demonstrează receptivitate pentru contexte noi de învățare.

9. Metode de predare *(Se vor avea în vedere metode care să asigure predarea centrată pe student. Se va descrie modul în care se asigură participarea studenților la stabilirea propriului parcurs de învățare, cum se identifică eventualele rămâneri în urmă și ce măsuri remediale se adoptă în astfel de cazuri.)*

În cadrul acestei materii, considerând atât orele de curs cât şi cele de aplicaţii, se vor folosi metode de predare atât expozitive (prelegerea, expunerea), cât și conversative-interactive, bazate pe modele de învățare prin descoperire facilitate de explorarea directa și indirectă a realității (experimentul, demonstrația, modelarea), dar și pe metode bazate pe acțiune, precum exercițiul, activitățile practice și rezolvarea de probleme.
În activitatea de predare vor fi utilizate prelegeri, în baza unor prezentări Power Point. Prezentările utilizează imagini și scheme, astfel încât informațiile prezentate să fie ușor de înțeles și asimilat.
Prezentările pot fi realizate şi cu ajutorul canalelor de comunicaţii online, pe platforma Teams.
Se va avea în vedere exersarea abilităţilor de ascultare activă şi de comunicare asertivă, precum şi a mecanismelor de construcţie a feedback-ului. Se va exersa abilitatea de lucru în echipă pentru rezolvarea diferitelor sarcini de învățare.

10. Conţinuturi

CURS		
Capitolul	**Conţinutul**	**Nr. ore**
1	Limbaje de programare 1.1. Prezentarea limbajelor de programare 1.2. Particularitati, avantaje, platforme/biblioteci de dezvoltare	2
2	Limbaje de programare 2.1. Particularitati, avantaje, platforme/biblioteci de dezvoltare 2.2. C/C++, Java, #c, Python	2
3	Arhitecturi pentru procesarea informatiei 3.1. Prezentarea arhitecturilor de calcul 3.2. Multiprocesoare, cloud, GPU	2
4	Arhitecturi GPU 4.1. Arhitecturi GPGPU, GPU 4.2. Arhitectura CUDA	2

5	Arhitectura CUDA 5.1. Istoric, evolutie 5.2. Descriere 5.3. Concepte specifice CUDA 5.4. Modele de programare CUDA 5.5. Patternuri de programare. Paralelism la nivel de date. Tipurti de memorie CUDA. 5.6. Aplicatii. Analiza si optimizare. Performanta.	5
6	Biblioteci bazate pe CUDA 6.1. Prezentarea bibliotecilor bazate pe arhitectura CUDA 6.2. cuBLAS, OpenACC,OpenCL, DNN	4
7	Aplicatii 7.1. Aplicatii ale prelucrarii informatiei cu implementari GPU/CUDA: prelucrarea imaginilor, inteligenta artificiala, computer vision, grafica 3D	4
	Total:	21

Bibliografie:

David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors 2nd edition - A Hands-on Approach, Morgan Kaufmann 2013
nVidia CUDA: www.developer.nvidia.com
Matt Pharr , Randima Fernando : GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation, Pearson Education 2005,
http://download.nvidia.com/developer/GPU_Gems_2/CD/Index.html
Jason sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison Wesley 2010,
http://www.mat.unimi.it/users/sansotte/cuda/CUDA_by_Example.pdf

LABORATOR

Nr. crt.	Conţinutul	Nr. ore
1	Biblioteci GPU/CUDA.	2
2	Dezvoltarea unei aplicatii CUDA	2
3	Dezvoltarea unei aplicatii CUDA	2
4	Analiza/profilingul aplicatiilor CUDA	2
5	Optimizare	2
6	Aplicatii: prelucrarea imaginilor	2
7	Aplicatii: prelucrarea imaginilor	2
	Total:	14

Bibliografie:

CUDA C Programming Guide, http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html
CUDA C Best Practices Guide, http://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html
David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors 2nd edition - A Hands-on Approach, Morgan Kaufmann 2013
Jason sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison Wesley 2010,
http://www.mat.unimi.it/users/sansotte/cuda/CUDA_by_Example.pdf

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare	11.3 Pondere din nota finală
11.4 Curs	- cunoaşterea noţiunilor teoretice fundamentale; - cunoaşterea tehnicilor de programare si a arhitecturilor; - cunoasterea platgormei GPU/CUDA; - cunoasterea unei biblioteci bazata pe GPU/CUDA	Teste oral, susţinute la date fixate in sesiune; subiectele acoperă întreaga materie, realizând o sinteză între parcurgerea teoretică comparativă a materiei şi explicitarea prin exerciţii a tehnicilor specifice programarii platformelor GPU.	50%
11.5 Seminar/laborator/proiect	- cunoaşterea modului de proiectare a unui algoritm de analiza a imaginilor pentru rezolvarea unei probleme date; - cunoaşterea modului de transpunere în cod a unui algoritm; - demonstrarea funcţionării unui algoritm implementat.	Proiectul final de laborator, cuprinzând o componentă teoretică şi o componentă practică. Componenta teoretică este verificată prin elaborarea unui referat; componenta practică este evaluată prin verificarea modului de rezolvare (implementare, testare, funcţionare) de către student a unor probleme practice.	50%
11.6 Condiţii de promovare			
Proiectarea, implementarea, şi demonstrarea funcţionării unor soluţii pentru procesarea imaginilor, computer vision, inteligenta artificiala folosind biblioteci cu accelerare GPU/CUDA. Obţinerea a 50% din punctajul total. Obţinerea a 50% din punctajul aferent activităţii de laborator pe parcursul semestrului.			

12. Coroborarea conţinutului disciplinei cu aşteptările reprezentanţilor angajatorilor şi asociaţiilor profesionale reprezentative din domeniul aferent programului, precum şi cu stadiul actual al cunoaşterii în domeniul ştiinţific abordat şi practicile în instituţii de învăţământ superior din Spaţiul European al Învăţământului Superior (SEÎS)

Procesarea informatiei necesita cunoasterea arhitecturilor disponibile in prezent pe piata, a limbajelor de programare, a structurilor de date necesare precum si a bibliotecilor existente adaptate diverselor domenii precum prelucrarea imaginilor, computer vision, inteligenta artificiala, grafica 3D, calcule matematice, algebra. Intelegerea acestor aspecte impreuna cu particularitatile fiecarei arhitecturi hardware creeaza premizele unor implementari eficiente, optimizate si adaptate cerintelor de calcul din ce in ce mai crescute odata cu cresterea complexitatii aplicatiilor. Prelucrarea rapida a informatiei necesita utilizarea unor platforme adaptate, fie ca se utilizeaza arhitecturi cloud, arhitecturi GPU sau platforme mobile. Procesarea informatiei necesita cunoasterea arhitecturilor disponibile in prezent pe piata, a limbajelor de programare, a structurilor de date necesare precum si a bibliotecilor existente adaptate diverselor domenii precum prelucrarea imaginilor, computer vision, inteligenta artificiala, grafica 3D, calcule matematice, algebra.

Intelegerea acestor aspecte impreuna cu particularitatile fiecarei arhitecturi hardware creeaza premizele unor implementari eficiente, optimizate si adaptate cerintelor de calcul din ce in ce mai crescute odata cu cresterea complexitatii aplicatiilor. Prelucrarea rapida a informatiei necesita utilizarea unor platforme adaptate, fie ca se utilizeaza arhitecturi cloud, arhitecturi GPU sau platforme mobile.

Data completării	Titular de curs	Titular(i) de aplicaţii
21.09.2026	S.l./Lect. Dr. George Valentin STOICA	S.l./Lect. Dr. George Valentin STOICA

Data avizării în departament	Director de departament
	Conf.Dr.Ing. Bogdan Cristian FLOREA

Data aprobării în Consiliul Facultăţii	Decan
22.09.2026	Prof. Dr. Mihnea Udrea